cocos-2dx

目录结构

build 编译

cocos cocos2d-x的库

doc 帮助文档

extensions 扩展的功能（工具）

external 扩展功能

licenses 授权文件

plugin 第三方的社交平台

templates 创建cpp项目模板文件

tests 测试的demo

tools 项目产生攻击等

# 项目安装

**安装mkvirtualenv -p python2目录 ENV27**

## 配置项目

（1）下载源码包

（2）解压

（3）安装必要程序

python：目前说要安装2.7版本

android sdk：vs提供

android ndk：

Ant：vs提供

IDE：

windows:安装vs2017 community

androde:安装java sdk

（4）运行setup.py

（5）重启cmd，运行命令，创建项目

// 创建项目-l 使用的语言

cocos new -p 名字 -d 保存的目录 -l cpp 项目名称

## 编译运行项目

每次编译只会编译差异文件

1. 进入项目文件
2. 运行cocos run -p win32 //win32是windows平台

## 开发

**windows：**

下使用vs2017打开\*\*.sln文件(sln项目文件）

通过vs2017运行调试

# 系统框架

## ApplicationDelegate|应用程序入口

**/classes/application**

// 启动程序--程序启动入口

ApplicationDidFinishLunching(){

绑定窗口，绑定导演

设置FPS

创建一个场景Scene

director->runWithScene()

}

// 进入后台程序

applicationDidEnterBackground(){

导演控制所以动作停止,停止音效

Director::getInstance()->stopAnimation();

}

// 返回再启动

applicationWillEnterForeground(){}

## Director |导演类

void runWithScene() 管理

void pushScene()

void popScene()

void popToRootScene()

void popToSceneStackLevel()

void replaceScence()

## Scene |图层类型

## Node |绘图节点和锚点

## AutoReleasePool |内存管理

使用C++实现了引用计数内存管理机制

所有对象都是Ref类的子类-Ref实现了管理机制

## Vector |集合

## Map|

## Layer|图层类

# 功能开发

1、

# 开发流程

1. 新建游戏原型Axure RP
2. 使用cocos2d命令创建一个新项目
3. 打开/classes/application.cpp,这里是入口文件，新建项目
4. 新建文件Classes中新建文件

# Hello World

1. Hellow.h

#include “cocos2d.h” //包含相关类的声明

USING\_NS\_CC; // cocos2d.h中定义的宏（命名空间）

class MainMenu::public Layer{

public:

CREATE\_FUNC(MainMenu); // 初始化方法

virtual bool init();

void menucallback(Ref \* object); // 回调函数

}